

Национальный исследовательский университет ИТМО

Факультет ПИиКТ

Лабораторная работа №4 по дисциплине
«Тестирование программного обеспечения»

Работу выполнили:

Асташин С. С. и Воробьев К. О.

Группа:

P33302

Преподаватель:

Харитонов А. Е.

Санкт-Петербург,

2023

Текст задания

С помощью программного пакета [Apache JMeter](#) провести нагрузочное и стресс-тестирование веб-приложения в соответствии с вариантом задания.

В ходе нагрузочного тестирования необходимо протестировать 3 конфигурации аппаратного обеспечения и выбрать среди них наиболее дешёвую, удовлетворяющую требованиям по максимальному времени отклика приложения при заданной нагрузке (в соответствии с вариантом).

В ходе стресс-тестирования необходимо определить, при какой нагрузке выбранная на предыдущем шаге конфигурация перестаёт удовлетворять требованиями по максимальному времени отклика. Для этого необходимо построить график зависимости времени отклика приложения от нагрузки.

Параметры тестируемого веб-приложения:

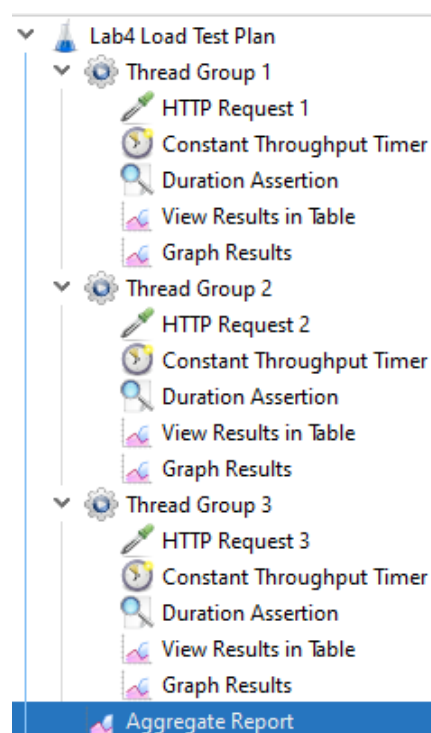
- URL первой конфигурации (\$ 2400) -
<http://stload.se.ifmo.ru:8080?token=468631640&user=2081020586&config=1>;
- URL второй конфигурации (\$ 4200) -
<http://stload.se.ifmo.ru:8080?token=468631640&user=2081020586&config=2>;
- URL третьей конфигурации (\$ 6100) -
<http://stload.se.ifmo.ru:8080?token=468631640&user=2081020586&config=3>;
- Максимальное количество параллельных пользователей - 7;
- Средняя нагрузка, формируемая одним пользователем - 40 запр. в мин.;
- Максимально допустимое время обработки запроса - 630 мс.

Исходный код

https://github.com/Gramdel/tpo_lab4/

Описание конфигурации JMeter для нагрузочного тестирования

План тестирования выглядит так:



Поскольку тред-группы отличаются только параметрами, которые передаются в GET-запрос, далее приведем скриншоты только первой из них:

The screenshot shows the 'Thread Group' configuration dialog. The 'Name' field is 'Thread Group 1'. The 'Comments' field is empty. The 'Action to be taken after a Sampler error' section has 'Continue' selected. The 'Thread Properties' section has 'Прямоугольник' selected. The 'Number of Threads (users)' is 7, 'Ramp-up period (seconds)' is 60, and 'Loop Count' is 40. The 'Same user on each iteration' checkbox is checked. The 'Delay Thread creation until needed' and 'Specify Thread lifetime' checkboxes are unchecked. The 'Duration (seconds)' and 'Startup delay (seconds)' fields are empty.

HTTP Request

Name:

HTTP Request 1

Comments:

Basic

Advanced

Web Server

Protocol [http]:

http

Server Name or IP:

localhost

Port Number:

8888

HTTP Request

GET

Path:

Content encoding:

☐ Redirect Automatically

☒ Follow Redirects

☒ Use KeepAlive

☐ Use multipart/form-data

☐ Browser-compatible headers

Parameters

Body Data

Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
token	468631640	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
user	2081020586	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
config	1	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

Constant Throughput Timer

Name:

Constant Throughput Timer

Comments:

Delay before each affected sampler

Target throughput (in samples per minute):

40.0

Calculate Throughput based on:

this thread only

Duration Assertion

Name:

Duration Assertion

Comments:

Apply to:

☐ Main sample and sub-samples

☒ Main sample only

☐ Sub-samples only

Duration to Assert

Duration in milliseconds:

630

View Results in Table

Name:

View Results in Table

Comments:

Write results to file / Read from file

Filename

1.csv

Browse...

Log/Display Only:

☐ Errors

☐ Successes

Configure

Sample #	1	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	14:54:57.024	Thread Group 1 1-1	HTTP Request 1	957		231	158	957	1	
2	14:54:58.524	Thread Group 1 1-1	HTTP Request 1	927		231	158	927	0	
3	14:55:00.024	Thread Group 1 1-1	HTTP Request 1	925		231	158	925	0	
4	14:55:01.524	Thread Group 1 1-1	HTTP Request 1	925		231	158	925	0	
5	14:55:03.024	Thread Group 1 1-1	HTTP Request 1	926		231	158	926	0	
6	14:55:04.524	Thread Group 1 1-1	HTTP Request 1	928		231	158	928	0	
7	14:55:05.595	Thread Group 1 1-2	HTTP Request 1	960		231	158	960	0	
8	14:55:06.024	Thread Group 1 1-1	HTTP Request 1	929		231	158	929	0	
9	14:55:07.096	Thread Group 1 1-2	HTTP Request 1	932		231	158	932	0	
10	14:55:07.524	Thread Group 1 1-1	HTTP Request 1	927		231	158	927	0	
11	14:55:08.595	Thread Group 1 1-2	HTTP Request 1	928		231	158	928	0	
12	14:55:09.023	Thread Group 1 1-1	HTTP Request 1	931		231	158	931	0	
13	14:55:10.095	Thread Group 1 1-2	HTTP Request 1	927		231	158	927	0	
14	14:55:10.524	Thread Group 1 1-1	HTTP Request 1	932		231	158	932	0	
15	14:55:11.595	Thread Group 1 1-2	HTTP Request 1	931		231	158	931	0	
16	14:55:12.024	Thread Group 1 1-1	HTTP Request 1	932		231	158	932	0	
17	14:55:13.094	Thread Group 1 1-2	HTTP Request 1	929		231	158	929	0	
18	14:55:13.524	Thread Group 1 1-1	HTTP Request 1	933		231	158	933	0	
19	14:55:14.167	Thread Group 1 1-3	HTTP Request 1	959		231	158	959	0	
20	14:55:14.595	Thread Group 1 1-2	HTTP Request 1	935		231	158	935	0	
21	14:55:15.024	Thread Group 1 1-1	HTTP Request 1	934		231	158	934	0	
22	14:55:15.667	Thread Group 1 1-3	HTTP Request 1	936		231	158	936	0	
23	14:55:16.095	Thread Group 1 1-2	HTTP Request 1	935		231	158	935	0	
24	14:55:16.524	Thread Group 1 1-1	HTTP Request 1	937		231	158	937	0	
25	14:55:17.167	Thread Group 1 1-3	HTTP Request 1	929		231	158	929	0	
26	14:55:17.595	Thread Group 1 1-2	HTTP Request 1	932		231	158	932	0	
27	14:55:18.024	Thread Group 1 1-1	HTTP Request 1	936		231	158	936	0	
28	14:55:18.666	Thread Group 1 1-3	HTTP Request 1	938		231	158	938	0	
29	14:55:19.094	Thread Group 1 1-2	HTTP Request 1	933		231	158	933	0	
30	14:55:19.524	Thread Group 1 1-1	HTTP Request 1	935		231	158	935	0	
31	14:55:20.166	Thread Group 1 1-3	HTTP Request 1	938		231	158	938	0	
32	14:55:20.595	Thread Group 1 1-2	HTTP Request 1	937		231	158	937	0	
33	14:55:21.024	Thread Group 1 1-1	HTTP Request 1	936		231	158	936	0	
34	14:55:21.666	Thread Group 1 1-3	HTTP Request 1	940		231	158	940	0	
35	14:55:22.095	Thread Group 1 1-2	HTTP Request 1	938		231	158	938	0	
36	14:55:22.523	Thread Group 1 1-1	HTTP Request 1	934		231	158	934	0	

☒ Scroll automatically?

☐ Child samples?

No of Samples 280

Latest Sample 929

Average 947

Deviation 13

Aggregate Report

Name:

Aggregate Report

Comments:

Write results to file / Read from file

Filename

agr.csv

Browse...

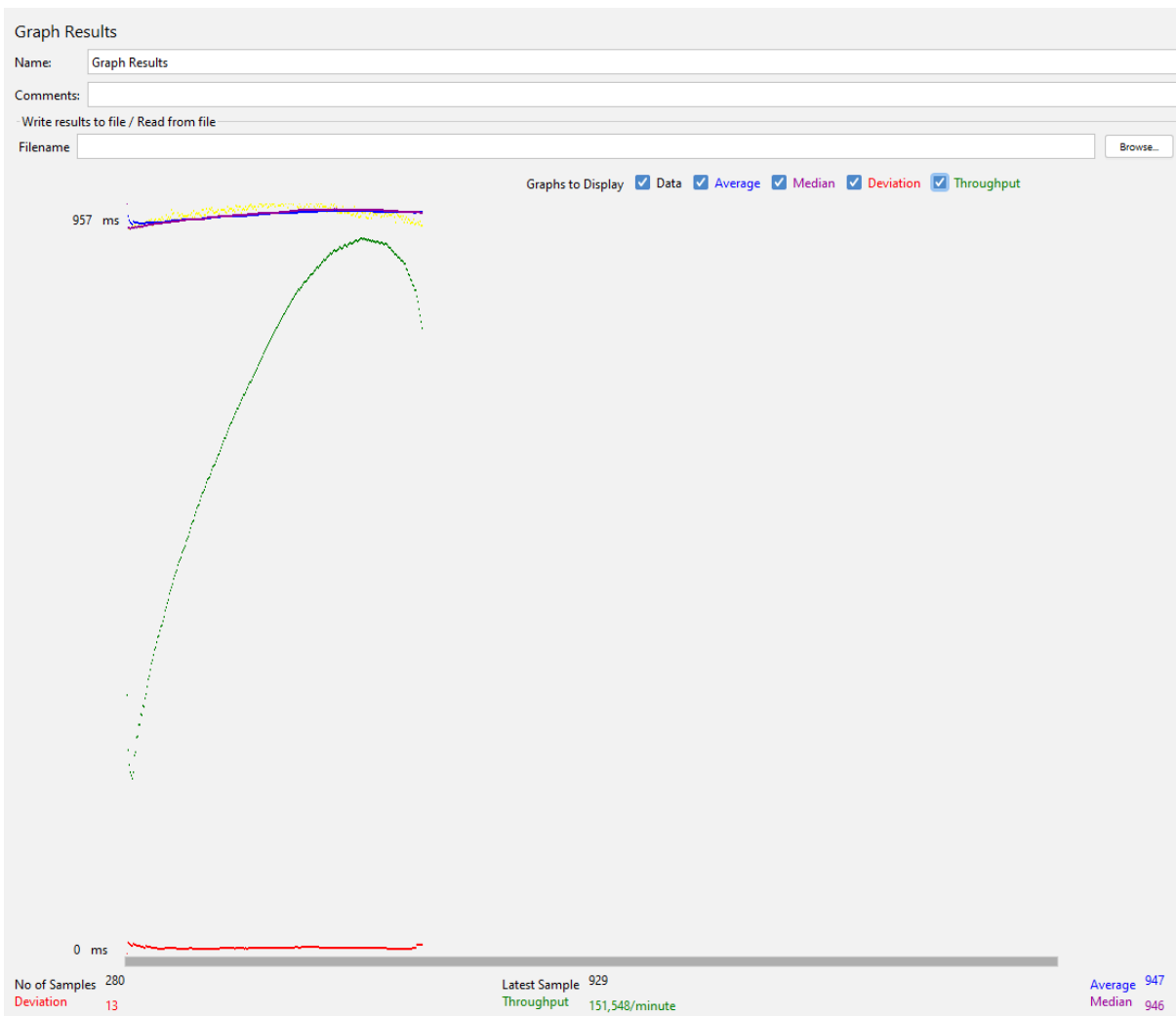
Log/Display Only:

☐ Errors

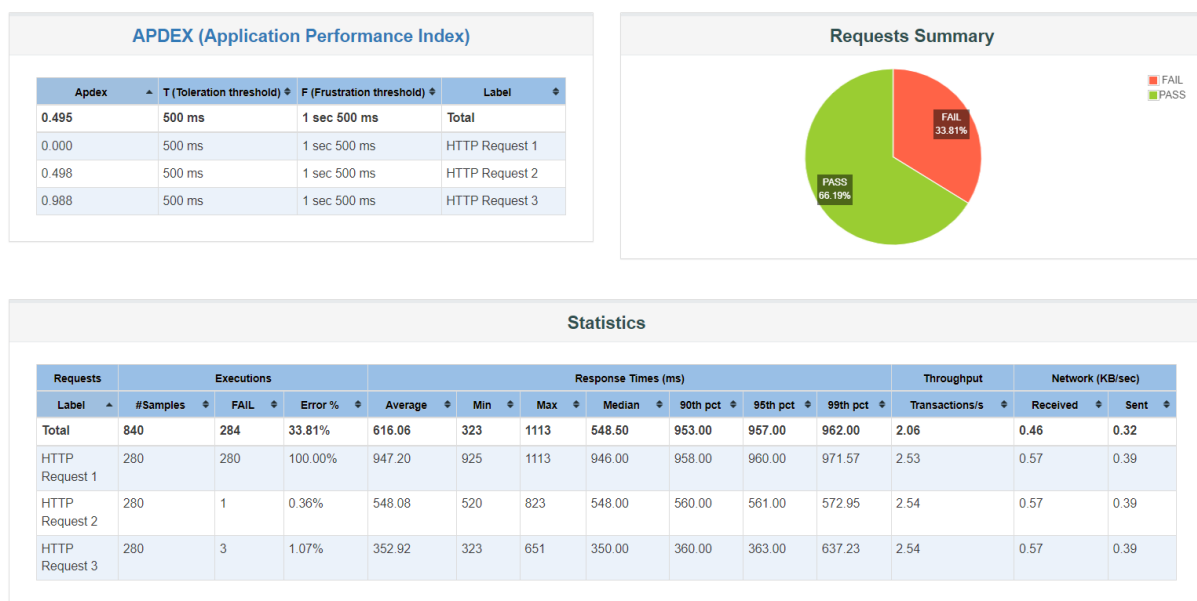
☐ Successes

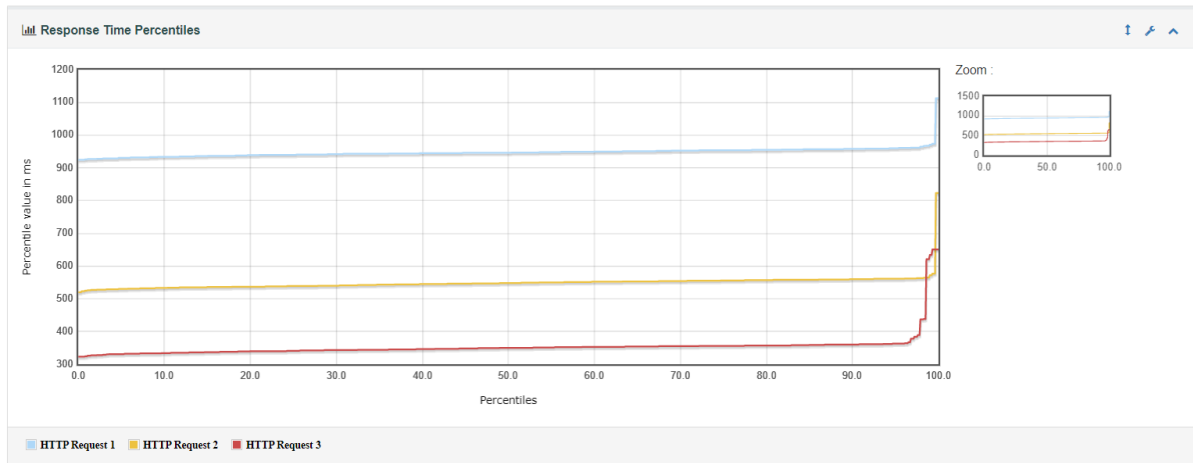
Configure

Label	1	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
HTTP Request 1		280	947	946	958	960	968	925	1113	100.00%	2,5/sec	0,57	0,39
HTTP Request 2		280	548	548	560	561	565	520	823	0,36%	2,5/sec	0,57	0,39
HTTP Request 3		280	352	350	360	363	621	323	651	1,07%	2,5/sec	0,57	0,39
TOTAL		840	616	548	953	957	962	323	1113	33,81%	2,1/sec	0,46	0,32



Выводы по результатам нагрузочного тестирования



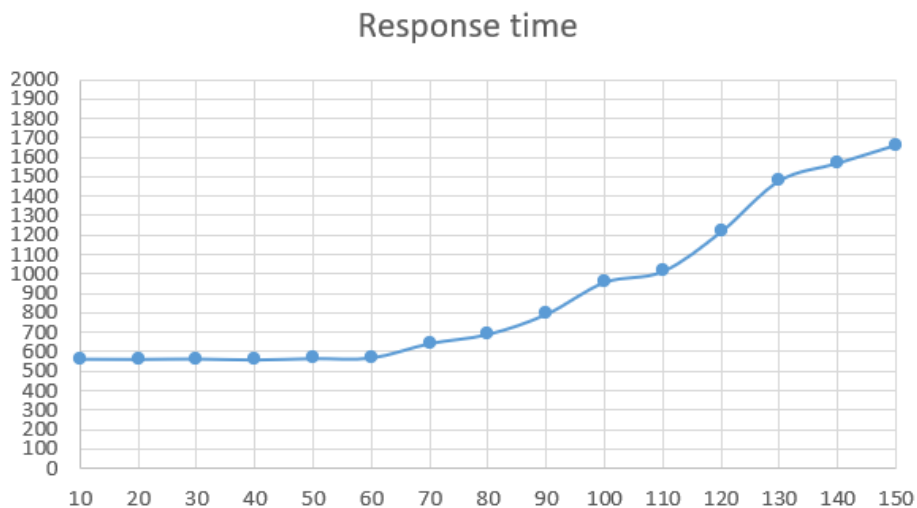


Исходя из результатов нагрузочного тестирования, можно увидеть, что 1-я конфигурация совсем не соответствует требованиям (все запросы выполняются дольше заданного максимального времени, среднее время 947.2 мс). При 2-й и 3-й конфигурации почти все запросы (более 99%) не превышают ограничение, причем процент ошибок и цена меньше у 2-й конфигурации.

Очевидно, что оптимальная конфигурация - вторая, поскольку она дешевле третьей, хотя 99% запросов выполнены за 572.95 мс (что на грани, т.к. пороговое значение - 630 мс) и с точки зрения надежности, возможно, третья конфигурация проявит себя лучше.

Описание конфигурации JMeter для стресс-тестирования

В ходе стресс-тестирования одновременно запускалось разное кол-во thread'ов (от 10 и до 150). До 60 одновременно запущенных thread'ов время ответа особо не изменяется, но при бОльшем значении уже начинает расти. При 70 одновременно запущенных thread'ах время обработки запроса приближается к максимальному допустимому значению по варианту (630 мс), и после 70 начинаются проблемы (время превышает, хотя сервер продолжает работать).



Вывод

В ходе данной лабораторной работы было проведено нагрузочное и стресс-тестирование сервера, чтобы выбрать оптимальную (с точки зрения стоимости) конфигурацию из представленных в варианте. По итогам нагрузочного тестирования была выбрана вторая конфигурация, поскольку она удовлетворяет требованиям и при этом имеет наименьшую стоимость. Также, было проведено стресс-тестирование выбранной конфигурации, по результатам которого выявлено, что она перестает удовлетворять требованиям по максимальному времени отклика при 70+ одновременно запущенных thread'ах.